Integritní omezení

Integrita dat

- **Integrita dat** = fakt, že data věrně zobrazují realitu, kterou popisují:
  - Jsou aktuální, databáze neobsahuje již neplatná data (kromě archivace historických dat),
  - je dodržena **referenční integrita**: Související informace v různých tabulkách si vždy odpovídají, provázaná data jsou aktuální, nechybí část souvisejících informací (při mazání musíme dbát na to, aby byly odmazány všechny související informace z různých tabulek).
- Integrita dat se zajišťuje vytvářením omezení, která klademe na data.

Integritní omezení

- Referenční integrita se zajišťuje vytvářením integritních omezení.
- **Integritní omezení** jsou samostatné objekty v databázi, které kladou podmínky na data vkládaná do určitého sloupce.
- **Realizace integritních omezení** v relačních databázích:
  - datové typy atributů – omezení typu, délky dat,
  - primární a cizí klíče – kontrola správnosti vazeb,
  - unique – zajištění jedinečnosti hodnot ve sloupci,
  - not null – data ve sloupci musí být vždy vyplněna,
  - check – ověření splnění podmínky,
  - triggery – pokročilé kontroly dat pomocí PL/SQL.

Kontrola integrity v databázi

- Integritní omezení zvyšují časové nároky dotazů, ale vždy se je vyplatí implementovat kvůli snadnému zajištění integrity. Je lepší *data kontrolovat na úrovni databáze namísto kontroly v aplikaci*, protože:
  - nad jedinou databází obvykle vzniká více aplikací,
  - kontrola na straně aplikace je zpravidla časově náročnější a implementuje se obtížněji,
  - úpravy integritních omezení lze provést na jediném místě – v databázi,
  - databázový systém je více samodokumentujícím.

Kontrola integrity v aplikacích

- Na straně aplikace lze provést dodatečnou (doplňkovou) kontrolu:
  - U webových aplikací JavaScriptem na straně klienta, obdobně u desktopových aplikací:
    - Zlepšení dojmu z uživatelského rozhraní,
    - zavedení filtrů, dynamické nápovědy,
    - kontrola dat odesílaných ve formulářích zabrání zbytečným požadavkům na server.

Integritní omezení pro sloupce

- **NOT NULL** – ve sloupci nesmí být speciální hodnota NULL udávající, že data nejsou vyplněna (tj. ve sloupci musí být vždy nějaká hodnota)
- **UNIQUE** – hodnoty ve sloupci musí být jedinečné
- **PRIMARY KEY** – označení primárního klíče = automaticky se přiřadí také UNIQUE a NOT NULL
- **FOREIGN KEY** – označení cizího klíče – sloupce, který odkazuje na sloupec, který je primárním klíčem v jiné tabulce
- **CHECK** – kontrola podmínky, kterou data musí splňovat, např. CHECK (ID BETWEEN 0 AND 100). Téměř všechny relační databáze mají CHECK (snad jenom kromě SAP HANA)
- **DEFAULT** – výchozí hodnota použitá pro sloupec pokud při vkládání dat žádnou neuvedeme

Pojmenovaná integritní omezení

- Integritní omezení lze volitelně pojmenovat (uvést nepovinné CONSTRAINT <název omezení>). Pokud to neprovedete, tak Oracle vygeneruje název omezení začínající s předponou SYS_, s takovými omezeními se ale hůř pracuje.
- Stávající omezení můžete upravovat pomocí příkazu ALTER (vypnout nebo zapnout).
- Dočasné vypnutí integritních omezení může být vhodné při importu dat z jiné databáze – data se například vkládají v blíže neurčeném pořadí, mohlo by dojít k dočasnému porušení omezení pro cizí klíče.
- Integritní omezení nepůjdou znovu zapnout, dokud data nebudou splňovat příslušné požadavky (je potřeba provést nezbytné úpravy dat, pokud nejsou zcela správná, a potom omezení aktivovat).

Změny integritních omezení

- Vypnutí/zapnutí integritního omezení:
  - ALTER TABLE employees_demo {DISABLE|ENABLE} CONSTRAINT empd_salary_min;
- Zrušení integritního omezení:
  - ALTER TABLE employees_demo DROP CONSTRAINT empd_salary_min;
- Přidání integritního omezení (jméno omezení nesmí už existovat):
  - ALTER TABLE employees_demo ADD CONSTRAINT empd_salary_min CHECK (salary > 8000);
- Předefinování sloupce (jména omezení nesmí už existovat):
  - ALTER TABLE employees_demo MODIFY (salary NUMBER(8,2) CONSTRAINT empd_salary_nn NOT NULL CONSTRAINT empd_salary_min CHECK (salary > 8000));

ON DELETE CASCADE / SET NULL

- Když vytvoříte cizí klíč, tak nemůžete smazat záznamy, které jsou obsažené v jiné tabulce. Toto je velice důležitý způsob zachování referenční integrity.
- Abyste takové záznamy smazali, tak je musíte smazat v cizích tabulkách a poté když nejsou nikde použity je můžete smazat ve zdrojové tabulce.
- Toto omezení je někdy nevyhovující, proto je možné na cizím klíči nastavit:
  - `ON DELETE CASCADE` – při mazání se nejprve automaticky smažou záznamy v cizí tabulce.
  - `ON DELETE SET NULL` – při mazání se v cizí tabulce nastaví hodnota cizího klíče na NULL.

Příkaz DROP TABLE

- Odstraní z databáze tabulku s daným jménem
  - DROP TABLE <název tabulky>;
- Tabulka bude odstraněna i s případnými indexy (viz. dále), které jsou pro ni definovány a se všemi integritními omezeními.
- Bude zrušena struktura tabulky, a tudíž i všechna data v tabulce!